Viewing and Projections
What are Projections?
Classical Projections

- Front elevation
- Elevation oblique
- Plan oblique
- Isometric
- One-point perspective
- Three-point perspective
Planar Geometric Projections

- Standard projections project onto a plane
- Projectors are lines that either:
  - Converge at center of projection
  - Are parallel
- Preserve lines but not angles
Remember Art Class?
Projection Taxonomy

planar geometric projections

parallel

multiview orthographic

axonometric

isometric dimetric trimetric

perspective

1 point 2 point 3 point

oblique
Orthographic Projection

Projectors orthogonal to projection surface
Orthographic Uses

Preserves shape and measurements (great for CAD)

Need isometric to see what’s hidden
Default Camera Projection

Orthographic is default

\[ \mathbf{p}_p = \mathbf{M} \mathbf{p} \]

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
x_p = x \\
y_p = y \\
z_p = 0 \\
w_p = 1
\]
Projecting onto a Screen

Define area of screen and clip coordinates

glOrtho(left, right, bottom, top, near, far)
Normalized Device Coordinates

Transformed clipped coordinates to normalized device coordinates (NDC)

`glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);`

(coordinates outside NDC discarded)
Why Use NDC?

Provides a standard range for plotting onto a device/screen

“Screen space” coordinates that can then be transformed into device coordinates
Orthographic Eye to NDC

• Move center to origin
• Scale to have sides of length 2

\[
\begin{bmatrix}
\frac{2}{\text{right} - \text{left}} & 0 & 0 & -\frac{\text{right} + \text{left}}{\text{right} - \text{left}} \\
0 & \frac{2}{\text{top} - \text{bottom}} & 0 & -\frac{\text{top} + \text{bottom}}{\text{top} - \text{bottom}} \\
0 & 0 & \frac{2}{\text{near} - \text{far}} & -\frac{\text{far} + \text{near}}{\text{far} - \text{near}} \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
Perspective Projection

- Converge at point along projection (vanishing point)
- Multiple vanishing points in multi-point perspective
Projective Space

- $w$ provides extra dimension to $(x, y, z)$ coordinate space
- Acts as a scaling value to represent distance from projector
  - Larger $w$ values correspond to more distance from viewer
Simple Perspective

- Center of projection at origin
- $z$ is projection plane

$x_p = \frac{x}{z/d}$

$y_p = \frac{y}{z/d}$

$z_p = d$
Homogeneous Form

consider \( \mathbf{M} \mathbf{p} = \mathbf{p}' \) where:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1/d & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= 
\begin{bmatrix}
x \\
y \\
z \\
z/d
\end{bmatrix}
\]

Apply perspective division (convert coordinate back to \( w=1 \)) to be NDC 
\( \mathbf{p}' = (dx/z, dy/z, d, 1) \)
Perspective Projection

`glFrustum(left, right, bottom, top, near, far)`
Projecting onto the Near Plane

Map eye space point \((x_e, y_e, z_e)\) to near plane point \((x_p, y_p, z_p)\)
Perspective Normalization

\[ z = -f_0 r \]

\[ (-1, -1, -1) \]

\[ (1, 1, -1) \]
Clipping

Clipping matrix:

\[
N = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & \alpha & \beta \\
0 & 0 & -1 & 0
\end{bmatrix}
\]

Normalized device coordinates:

\[
z_{ndc} = \frac{z_c}{w_c} = \frac{\alpha z_e + \beta w_e}{-z_e}
\]

\[
\alpha = \frac{\text{near} + \text{far}}{\text{far} - \text{near}}
\]

\[
\beta = \frac{2\text{near} \times \text{far}}{\text{near} - \text{far}}
\]

Near plane is mapped to \( z = -1 \)

Far plane is mapped to \( z = 1 \)

Sides are mapped to \( x = \pm 1, \ y = \pm 1 \)
General Frustum Transform

\[
\begin{bmatrix}
\frac{2n}{r - l} & 0 & \frac{r + l}{r - l} & 0 \\
0 & 2n & \frac{r - l}{t + b} & 0 \\
0 & \frac{t - b}{r - l} & \frac{t - b}{t - b} & 0 \\
0 & 0 & -\left(\frac{f + n}{f - n}\right) & -2fn \\
0 & 0 & \frac{f - n}{f - n} & 0 \\
0 & 0 & -1 & 0
\end{bmatrix}
\]
Symmetric Viewing Volume

When right = -left and top = -bottom:

\[ r + l = 0 \]
\[ r - l = 2r \]
\[ t + b = 0 \]
\[ t - b = 2t \]

\[
\begin{pmatrix}
\frac{n}{r} & 0 & 0 & 0 \\
0 & \frac{n}{t} & 0 & 0 \\
0 & 0 & -\frac{(f+n)}{f-n} & -\frac{2fn}{f-n} \\
0 & 0 & -1 & 0
\end{pmatrix}
\]
Note about Deprecation

glOrtho and glFrustum are deprecated as of OpenGL 3.0

Replacements:
glm::glOrtho
glm::glFrustum
In-Class Exercise

- Consider `glFrustum(-4, 4, -3, 3, 5, 80)` and `glOrtho(-4, 4, -3, 3, 5, 80)`
- Construct these projection matrices
- Apply these matrices to points:
  - p1 = (3, 2, 20)
  - p2 = (3, 2, 3)
  - p3 = (-2, -4, 10)
Additional Reading

http://www.songho.ca/opengl/gl_projectionmatrix.html